

patent application. In this event, the Commissioner is also authorized to charge Deposit Account No. 23-1665 for any fee that may be required to maintain the pendency of this patent application.

Please amend the application as follows.

IN THE CLAIMS

Please amend the claims to read as follows.

- sd
cl
1. (Twice Amended) A data processing system stored on a computer-readable medium comprising:
- a plurality of event modules each including code that generates an event data signal representative of a particular event;
 - a plurality of scripts each having a plurality of instructions that provide results;
 - a plurality of processing modules each including code that provides processed data to said plurality of scripts; and
 - a task module, selectively communicating with each of said plurality of event modules, said task module including code for executing a selected one of said plurality of scripts that corresponds to said event data signal, for executing said plurality of instructions within said selected script, and for selectively invoking at least one of said plurality of processing modules in accordance with information regarding statuses of said plurality of processing modules and results generated by previously executed instructions;
- wherein during execution of said selected script, said selected script incorporates said processed data to modify execution of next instructions of said selected script.
- B

2. (Twice Amended) The system as claimed in claim 1 wherein said task module executes two or more of said plurality of scripts substantially simultaneously.

3. (Amended) The system as claimed in claim 2 further comprising:
a converter module, in communication with said task module, including code that maps said event data signal to at least one of said two or more of said plurality of scripts upon reception of said event data signal by said task module.

4. (Amended) The system as claimed in claim 1 wherein said plurality of processing modules provide event data signals, representative of particular events, to said task module.

5. (Amended) The system as claimed in claim 1 further comprising:
a status monitoring module, in communication with said task module, including code that provides said status information to said task module including operating conditions of said plurality of processing modules.

6. (Amended) The system as claimed in claim 5 wherein said status monitoring module is in direct communication with said plurality of processing modules.

7. The system as claimed in claim 5 wherein during said execution of said selected script, said status monitoring module stores data associated with said selected script in an associated memory.

8. (Amended) The system as claimed in claim 1 further comprising:
a load balancing module, in communication with said task module, including code that dynamically selects available ones of said plurality of processing modules to perform processing.
9. (Amended) The system as claimed in claim 8 wherein said load balancing module is in direct communication with said plurality of processing modules.
10. (Amended) The system as claimed in claim 1 wherein said task module interfaces with said plurality of processing modules for bi-directionally and substantially simultaneously transmitting data between said plurality of processing modules and said task module.
11. (Amended) The system as claimed in claim 1 further comprising:
a resource management module, in communication with said task module, including code that dynamically assigns processing functions to said plurality of processing modules.
12. (Amended) The system as claimed in claim 11 wherein said resource management module is in direct communication with said plurality of processing modules.
13. (Amended) The system as claimed in claim 1 further comprising:
a plurality of initiator modules including code that provides a communication interface between an associated one of said plurality of processing modules and said task module.

14. (Amended) The system as claimed in claim 13 wherein each of said plurality of initiator modules communicates with said associated one of said plurality of processing modules regardless of native applications contained on said associated one of said plurality of processing modules.

15. (Amended) The system as claimed in claim 13 further comprising:
a protocol disposed between each of said plurality of initiator modules and said task module for providing a communication interface therebetween.

3 *sch*
CS > 16. (Amended) The system as claimed in claim 13 further comprising:
a protocol disposed between each of said plurality of initiator modules and said associated one of said plurality of processing modules for providing a communication interface therebetween.

pk 17. (Amended) The system as claimed in claim 1 further comprising:
a plurality of client modules including code that provides a communication interface between an associated one of said plurality of event modules and said task module.

18. (Amended) The system as described in claim 17 further comprising:
a protocol disposed between said task module and each of said plurality of client modules for providing a communication interface therebetween.

19. (Amended) The system as claimed in claim 17 further comprising:

a protocol disposed between each of said plurality of client modules and said associated one of said plurality of event modules for providing a communication interface therebetween.

20. (Amended) The system as claimed in claim 1 wherein each of said plurality of scripts is preprogrammed to iteratively update its contents.

21. The system as claimed in claim 1 further comprising:
a storage module, in communication with said task module, for providing storage for said system.

22. The system as claimed in claim 21 wherein said storage module comprises a computer-readable medium.

23. The system as claimed in claim 22 wherein said computer readable medium comprises a persistent memory.

24. (Amended) The system as claimed in claim 21 further comprising:
a script building module, in communication with said storage module, including code that creates said plurality of scripts.

25. The system as claimed in claim 24 wherein said script building module includes a standard language interface.

26. The system as claimed in claim 24 wherein said script building module includes a graphical user interface.

27. (Amended) The system as claimed in claim 24 wherein said script building module dynamically updates and modifies said plurality of scripts.

28. (Amended) The system as claimed in claim 1 further comprising:
a protocol for providing a communication interface between said task module and each of said plurality of event modules.

29. (Amended) The system as claimed in claim 1 further comprising:
a protocol for providing a communication interface between said task module and each of said plurality of processing modules.

30. (Twice Amended) The system as claimed in claim 1 further comprising:
a responder module, in communication with said task module, including code that transmits response data, resulting from said execution, from said task module in a particular format to said plurality of processing modules or in a particular format to said plurality of event modules.

31. (Amended) The system as claimed in claim 1 further comprising:
an administrative module, in communication with said task module, including code that receives and presents data that relates to said plurality of processing modules.

32. (Amended) The system as claimed in claim 1 further comprising:

a plurality of application peripherals in communication with an associated one of said plurality of processing modules or an associated one of said plurality of event modules.

33. (Twice Amended) A data processing system stored on a computer readable-medium comprising:

a plurality of event modules each including code that generates an event data signal representative of a particular event;

a plurality of scripts each having a plurality of instructions;

a plurality of processing modules each including code that provides said plurality of scripts with processed data;

a task module, selectively communicating with each of said plurality of event modules, including code for executing a selected one of said plurality of scripts that correspond to said event data signal, for executing said plurality of instructions within said selected script, and for selectively invoking at least one of said plurality of process modules in accordance with information regarding a status of said plurality of processing modules and results generated by previously executed instructions;

wherein during execution of said selected script said selected script incorporates said processed data to modify execution of next instructions of said selected script;

a resource management module in direct communication with said plurality of event modules, said task module and said plurality of processing modules, including code that dynamically assigns processing functions to said plurality of processing modules; and

an administrative module in direct communication with said task module, said plurality of event modules, said plurality of processing modules and said resource management module, including code that receives and presents data relating to said plurality of processing modules.

34. (Twice Amended) A method of data processing comprising the steps of:
generating at least one event data signal at one or more peripheral modules;
mapping said at least one event data signal to a selected script chosen from one or more scripts, each said one or more scripts having one or more instructions for performing data gathering steps; executing said one or more instructions within said selected script; and
invoking, by a task module in accordance with status information, one or more processing modules to process data required by said selected script;
wherein during execution of said one or more instructions, said selected script dynamically incorporates data processed by said one or more processing modules to modify execution of next ones of said one or more instructions of said selected script.

35. The method as claimed in claim 34 wherein said one or more peripheral modules and said task module communicate via a communication interface.

36. The method as claimed in claim 34 further comprising the step of:
dynamically managing operating functions of said one or more peripheral modules.

37. The method as claimed in claim 34 further comprising the steps of:
producing response data signals as a result of said executing step; and

transmitting said response data signals from said task module to selected said one or more peripheral modules.

38. The method as claimed in claim 37 further comprising the step of:
translating said response data signals transmitted from said task module into a format that said selected said one or more peripheral modules recognize.

39. The method as claimed in claim 38 further comprising the step of:
storing said event data signals, said one or more scripts and said response data signals in a storage medium that is in communication with said task module.

40. The method as claimed in claim 39 wherein said storage medium is persistent.

41. The method as claimed in claim 34 further comprising the step of:
accessing a protocol to interface between said task module and selected said one or more peripheral modules.

sa
C" }
a
b
42. (Amended) The method as claimed in claim 34 further comprising the step of:
providing communication between said task module and each of said one or more peripheral modules such that said task module invokes only ones of said one or more peripheral modules available for performing processing operations.

43. The method as claimed in claim 34 wherein said executing step includes the step of:

interfacing with a plurality of said one or more peripheral modules substantially simultaneously.

44. The method as claimed in claim 34 wherein said executing step executes a plurality of said one or more scripts substantially simultaneously.

45. (Amended) The method as claimed in claim 34 wherein said execution of said one or more instructions dynamically incorporates data gathered in previously executed instructions.

46. The method as claimed in claim 34 further comprising the step of:
providing results of said executing step to an administrative module for presenting information relating to said one or more peripheral modules.

Please add the following new claims.

60. In a data processing system, a method for responding to event data, comprising:
receiving event data from a requesting one of a plurality of event modules;
mapping the event data to one of a plurality of scripts, the plurality of scripts including instructions for responding to event data;
executing the instructions within the mapped script to generate results;
during the execution of at least one instruction, selecting and invoking one of a plurality of processing modules available for providing processed data to the at least one instruction, the selecting and invoking in accordance with information regarding a status of the processing

modules and results generated by previously executed instructions, and modifying execution of next instructions within the mapped script in accordance with the processed data generated by the selected and invoked processing module;

building a response profile including the generated results; and

wherein when the instructions within the mapped script are completed, transmitting the response profile to the requesting one of the plurality of event modules.

61. The method as claimed in claim 60 wherein the generated results include event data.

62. The method as claimed in claim 60, comprising:
tracing execution of the instructions within the mapped script and processing of the invoked processing modules; and

wherein when a processing module fails, continuing execution of the mapped script and the processing of the invoked processing modules from a last traced instruction.

63. A data processing system stored on a computer-readable medium comprising:
a plurality of event modules each including code that generates a first event data signal representative of a first event;

a plurality of scripts each having a plurality of instructions that provide results;

a plurality of processing modules each including code that provides processed data, a subset of said plurality of processing modules having code that selectively generates a second event data signal representative of a second event; and